PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of

HALTMEYER, John M.

Application No. 10/076,948

Filed: February 19, 2002

For: THOROUGH OPERATION RESTRICTION

Group Art Unit: 2232

Examiner: ALMEIDA, Devin E.

APPEAL BRIEF

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Statutes

35 U.S.C. § 102(b)..9

Rules

37 C.F.R. §41.33..5

Cases

None cited

In accordance with the provisions of 37 C.F.R. §41.37, Appellant submits herewith its brief in support of the appeal of the final rejection of claims 1, 2, 4 and 9.

(i) Real Party in Interest

The Real Party in Interest in the present Appeal is Tricerat, Inc., the assignee, as evidenced by the assignment set forth at Reel 017498, Frame 0823.

(ii) Related Appeals and Interferences

There are no related appeals, interferences or judicial proceedings known to Appellant's legal representative.

(iii) Status of Claims

Claims 1, 2, 4, and 9 stand finally rejected by the Examiner as noted in the Final Action mailed September 1, 2009. The rejection of claims 1, 2, 4 and 9 is appealed.

(iv) Status of Amendments

An amendment to claim 9 was submitted December 8, 2009, subsequent to the Final Action, pursuant to Rule 41.33. The amendment is to render the claim language more definite and remove limitations for which there was not proper antecedent basis. At the time of filing the brief, it is not know whether the Amendment has been entered. In the discussion of the claims and the claim appendix, reference will be made to claim 9 as if the amendment has been entered.

(v) Summary of Claimed Subject Matter

In independent claim 1, the invention relates to a method for controlling the applications that a computer user may run on a multi-user system. (Page 8, lines 10-11) When a computer user logs on to the multi-user system in user mode, a security executable (mjolnir.exe) on the multi-user system is automatically used to create a list of authorized applications in a database of the multi-user system for the user. (Page 9, lines 4-5; Fig. 2, step 20; Fig. 3, steps 100 and 110; Appendix B, page 6, lines 15 *et seq*) A hook function (thor32.dll) is attached to all new applications. (Page 9, lines 2-3; Fig. 2, step 10; Fig. 3, step 140; Fig. 4, steps 320, 330, and 340; Appendix A, page 2, line 10) When a new application is started, the hook function sends a message including the ID and path of the new application to the security executable. (Page 9, lines 6-7; Fig. 2, step 30; Fig. 3, step 150; Fig. 4, step 350; Appendix A, page 1, lines 40 *et seq*) The message is received from the hook function at the security executable and correlated to the list of authorized applications to determine whether the new application is authorized for the user. (Page 9, lines 8-9; Fig. 2, step 40; Fig. 3, steps 160 and 170; Fig. 4, step 360; Appendix B, page 9, line 35) The message is answered by the security executable when the new application is authorized. (Page 9, lines 10-12; Fig. 2, step 50; Fig. 3, step 180; Fig. 4, step 370; Appendix A, page 2, lines 29-31) If the new application is not authorized, the new application is stopped. (Page 9, lines 13-14; Fig. 2, step 60; Fig. 3, setp 190; Fig. 4, step 380; Appendix B, page 12, line 9)

In independent claim 2, the system for performing the method of claim 1 is defined. Specifically, the system of claim 2 controls the applications that a computer user may run on a multi-user system. The system includes a security executable (mjolnir.exe) in user mode which creates a list of authorized applications for the computer user in a database of the multi-user system. (Page 8, lines 18-21; Page 9, line 20 to Page 10, line 7 Fig. 1, Executive; database) A hook function is automatically attached to all new applications in user mode when the computer user logs on to the multi-user system. (Page 9, lines 2-3; Page 10, lines 8-19; Page 11, lines 1-17; Fig. 1, Win32 Application, Win32 Subsystem) Means are provided for querying an ID of each new application (Page 9, lines 5-9; Fig. 1, Security Reference Monitor) and means are also provided for sending a message with the application ID and the path of the application being examined using the security executable (Page 11, lines 18-20; Fig. 1. Security Reference Monitor). The system further includes means for retrieving the ID of each new application (Page 9, lines 8-9; Page 12, lines 2-4; Fig. 1, Executive Services; Fig. 2, step 40; Fig. 3, steps 160 and 170; Fig. 4, step 360; Appendix B, page 9, line 35) and means for terminating each new application not identified on the list of allowed applications. (Page 9, lines 13-14; Page 12, lines 7-8; Fig. 1, Process Manager; Fig. 2, step 60; Fig. 3, step 190; Fig. 4, step 380; Appendix B, page 12, line 9) In addition, the system includes means for answering a message when the application is identified on the list of allowed applications. (Page 9, lines 10-12; Page 12, lines 5-6; Fig. 1, Process Manager; Fig. 2, step 50; Fig. 3, step 180; Fig. 4, step 370; Appendix A, page 2, lines 29-31)

In claim 4, the system of claim 2 includes a system dynamic link library (which is used to attach the hook function to the new applications.

In independent claim 9 (as amended on December 8, 2009), the invention relates to a method similar to claim 1 but for use on a network rather than a multi-user computer system. More particularly, the invention relates to a method for controlling the applications that computer users may run on a network environment. (Page 8, lines 10-11) When a computer user logs on to the network, a security executable (mjolnir.exe) on the network in user mode is used to create and maintain a list of authorized applications in a database of the network and IDs for each computer user. (Page 9, lines 4-5; Fig. 2, step 20; Fig. 3, steps 100 and 110; Appendix B, page 6, lines 15 *et seq*) A hook function (thor32.dll) is attached to all new applications. (Page 9, lines 2-3; Fig. 2, step 10; Fig. 3, step 140; Fig. 4, steps 320, 330, and 340; Appendix A, page 2, line 10) The new applications that are started with the hook function are monitored and the application ID for each application is determined. (Page 9, lines 6-7; Fig. 2, step 30; Fig. 3, step 150; Fig. 4, step 350; Appendix A, page 1, lines 40 *et seq*) The application ID from the hook function is received by the security executable. (Page 9, lines 10-12; Fig. 2, step 50; Fig. 3, step 180; Fig. 4, step 370; Appendix A, page 2, lines 29-31) A determination is made if the application ID of the started application is on the list of authorized applications.

(Page 9, lines 8-9; Page 12, lines 2-4; Fig. 2, step 40; Fig. 3, step 170; Fig. 4, step 360) If so, the application is allowed to continue. (Page 9, lines 10-12; Page 12, lines 5-6; Fig. 2, step 50; Fig. 3, step 180; Fig. 4, step 370) If not, the application is terminated. (Page 9, lines 13-14; Page 12, 7-8; Fig. 2, step 60; Fig. 3, step 190; Fig. 4, step 380)

The claimed invention allows an administrator to build a list of applications in the multi-user system or server which a particular user is authorized to access and does not rely on the administrator to disallow a particular application. Because the hook and security executable run in user mode, only the processes created by the users are validated. This decreases the processing requirements for marshaling user processes. Operating system processes are not affected by the inventive process. Data loss due to program failure is eliminated because there is no interference with system I/O.

(vi) Grounds of Rejection to be Reviewed on Appeal

The Examiner rejected claims 1, 2, 4, and 9 under 35 U.S.C. 102(b) as being anticipated by U.S. Patent 7,165,269 (Winneg et al).

(vii) Argument

Winneg et al discloses a method and system which securely executes an application on a computer system such that the user of the computer system can not access unauthorized content available on the system or view content accessible via the system, such as the Internet. It essentially is a blocking system, preventing a computer user from obtaining "outside" information when the computer is utilized in a testing environment. Normally, a computer user would have access to information stored on the computer or accessible via the computer when the user is taking a test online or otherwise via the computer. This is analogous to an open book exam. Winneg discloses a system and method for creating a "closed book exam" environment by denying the computer user access to information stored on the computer's hard drive or accessible via the computer. See Figs. 8 and 9 and the discussion thereof at column 9 line 59 through column 15, line 32. Winneg is not interactive in that there is no inquiry regarding allowed applications or information, no comparison of the inquiry ID to stored IDs, and no granting of access to allowed applications as recited in independent claims 1, 2 and 9. Rather, Winneg essentially creates a brick wall, denying access to the computer user to all programs other than those necessary to take the exam.

In contrast, the present invention is directed to a method and system in which a security executable in user mode is utilized to create a list of authorized applications in a database on the network ("multi-user system") for the user, with a hook function being attached to all new applications when the computer user logs onto the network. The hook function is used to send a message including the ID for each application. The system is somewhat interactive in that the list of authorized IDs is automatically queried when the user seeks to initiate an application to determine whether the ID for the application to be initiated is within the list of authorized applications. If so, the security executable "answers" the query by allowing the user to have access to the application. If not, the application to be initiated is terminated, whereby the user is prevented from running the application. By performing the operations in user mode, the network is not burdened.

It should be noted that all of these features are explicitly recited in the claims, The differences between the present invention and Winneg will be seen by considering elements of the claims.

Claims 1 and 9 include the limitations of (1) providing a security executable on a network or multi-user system, and (2) automatically attaching a hook function in user mode to all new applications and employing the hook function whenever a new application is started to send a message to the security executable in user mode, the message including the ID and path of the new application. The Examiner contends that Winneg satisfies these limitations. Regarding the first limitation, the Examiner points to column 4, lines 8-18 of Winneg and contends that it discloses "automatically using a security executable on the multi-user system in user mode to create a list of authorized application (sic) in a database of the multi-user system." The portion of Winneg cited by the Examiner relates to transforming a user's computer, such as a laptop, personal computer, or workstation of a computer lab into a secure application-executing environment. Thus, Winneg does not disclose or suggest **using a security executable,** nor does Winneg disclose using a security executable **on the system or network,** as opposed to on the user's computer in user mode, as recited in the independent claims.

Regarding the second limitation, the Examiner points to column 12, line 43 to column 13, line 2 as teaching "attaching a hook function in user mode to all new applications; employing the hook function whenever a new application is started to send a message to the security executable in user mode." This portion of Winneg actually describes "hooks" and "filters". More relevant are the paragraphs preceding and succeeding the portion relied on by the Examiner. At column 12, lines 32-36, Winneg states: "To assist in disabling one or more of the functions on the computer system that are capable of accessing unauthorized content and/or displaying unauthorized content to a user of the computer system, one or more programming hooks **may** be utilized." (emphasis added) Similar language is found at column 13, lines 3-6. Thus, Winneg does not require that hooks are used to disable functions on a computer system as in the claimed invention. More significantly, Winneg does not teach that hook functions are applied in user mode to all new applications as claimed. The claims further state that the message sent to the security executable includes the ID and path of the new application. The Examiner does not cite any portion of Winneg for teaching this limitation. Rather, the Examiner improperly relies on a "SetWindowsHookEx" reference, Dietmoday inherent in windows to SetWindowsHookEX function parameter dwThreadID. This reference is not in the record and thus can not be relied on by the Examiner, particularly in a rejection based on Section 102.

Independent claim 2 recites similar limitations regarding a security executable on a multi-user system in user mode and a hook function which is automatically attached to all new applications in user mode when the computer user logs on to the system as are recited in claims 1 and 9. As set forth above, Winneg does not disclose these limitations.

Since Winneg does not teach each and every element of claims 1, 2, 4, and 9, the Applicant respectfully submits that these claims are not anticipated by Winneg.

Conclusion

In conclusion, the Examiner erred in rejecting claims 1, 2, 4, and 9 and should be reversed.

Please charge any government fees required for entry of this brief or credit any overpayment to Deposit Account 50-1936.

Respectfully submitted,

December 14, 2009

Lawrence E. Laubscher, Jr.
Registration No. 28,233
Laubscher & Laubscher, P.C.
1160 Spa Road
Suite 2B
Annapolis, MD 21403
Telephone: 410 280 6608

CERTIFICATE OF TRANSMISSION

I hereby certify that this correspondence is being transmitted electronically to: Mail Stop BPAI/MDE 9C33, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 via EFS-web on December 14, 2009.

Marianne G. Smith

(viii) Claims Appendix (with claim 9 as amended December 8, 2009)

1. A process for controlling the applications that a computer user may run on a multi-user system, comprising the steps of:

automatically using a security executable on the multi-user system in user mode to create a list of authorized applications in a database of the multi-user system for the computer user when the computer user logs on to the multi-user system;

attaching a hook function in user mode to all new applications;

employing the hook function whenever a new application is started to send a message to the security executable in user mode, said message including the ID and path of the new application;

receiving said message from the hook function at the security executable and correlating to said list to determine whether the new application is authorized;

answering the message by the security executable when the new application is authorized and;

stopping the new application when the new application is not authorized.

2. A system for controlling the applications that a computer user may run on a multi-user system, comprising:

a security executable in user mode for creating a list of authorized applications in a database of the multi-user system for the computer user;

a hook function which is automatically attached to all new applications in user mode when the computer user logs on to the multi-user system;

means for querying an ID of each said new application; and

means for sending a message with the application ID and the path of the application being examined using said security executable;

means for retrieving the ID of each new application;

means for terminating each new application not identified on said list of allowed applications; and means for answering a message when the application is identified on said list of allowed applications.

3. Cancelled.

4. The system for controlling the applications that a computer user may run according to claim 2, wherein said hook function is attached to said new applications by using a system dynamic link library.

5. Cancelled.

6. Cancelled.

7. Cancelled.

8. Cancelled.

9. A process for controlling the applications that computer users may run on a network, comprising the steps of:

using a security executable on the network in user mode to create and maintain a list of authorized applications in a database of the network and IDs for each computer user when the computer user logs on to the network;

attaching a hook function to all new applications;

monitoring all new applications that are started with the hook function and determining an application ID thereof;

receiving said application ID from the hook function by the security executable;

determining whether the application ID of each started application is on said list;

allowing said application to continue when its application ID is on the list; and

terminating said application when its application ID is not on the list.

(ix) Evidence Appendix

There is no exhibit for this appendix.

(x) Related Proceedings Appendix

There are no related proceedings, and accordingly no exhibit for this appendix.